Weighted Model Counting with Conditional Weights for Bayesian Networks

Paulius Dilkas Vaishak Belle

University of Edinburgh, Edinburgh, UK

UAI 2021

The Problem of Computing Probability

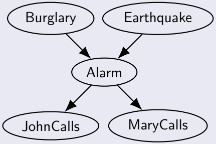
ProbLog

```
0.001 :: burglary.
0.002 :: earthquake.
0.95  :: alarm     :- burglary, earthquake.
0.94  :: alarm     :- burglary, \+ earthquake.
0.29  :: alarm     :- \+ burglary, earthquake.
0.001 :: alarm     :- \+ burglary, \+ earthquake.
0.9   :: johnCalls :- alarm.
0.05  :: johnCalls :- \+ alarm.
0.7   :: maryCalls :- alarm.
0.01  :: maryCalls :- \+ alarm.
```

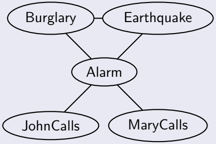
BLOG

```
random Boolean Burglary ~ BooleanDistrib(0.001);
random Boolean Earthquake ~ BooleanDistrib(0.002);
random Boolean Alarm ~
  if Burglary then
    if Earthquake then BooleanDistrib(0.95)
    else BooleanDistrib(0.94)
  else
    if Earthquake then BooleanDistrib(0.29)
    else BooleanDistrib(0.001);
random Boolean JohnCalls ~
  if Alarm then BooleanDistrib(0.9)
  else BooleanDistrib(0.05);
random Boolean MaryCalls ~
  if Alarm then BooleanDistrib(0.7)
  else BooleanDistrib(0.01);
```

Bayesian Network



Markov Random Field

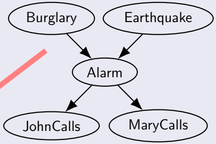


The Problem of Computing Probability

ProbLog

```
0.001 :: burglary.
0.002 :: earthquake.
0.95  :: alarm     :- burglary, earthquake.
0.94  :: alarm     :- burglary, \+ earthquake.
0.29  :: alarm     :- \+ burglary, earthquake.
0.001 :: alarm     :- \+ burglary, \+ earthquake.
0.9   :: johnCalls :- alarm.
0.05  :: johnCalls :- \+ alarm.
0.7   :: maryCalls :- alarm.
0.01  :: maryCalls :- \+ alarm.
```

Bayesian Network

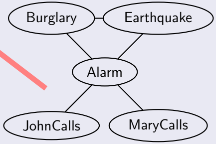


WMC

BLOG

```
random Boolean Burglary ~ BooleanDistr
random Boolean Earthquake ~ BooleanDi
random Boolean Alarm ~
  if Burglary then
    if Earthquake then BooleanDistrib(0.95)
    else BooleanDistrib(0.94)
  else
    if Earthquake then BooleanDistrib(0.29)
    else BooleanDistrib(0.001);
random Boolean JohnCalls ~
  if Alarm then BooleanDistrib(0.9)
  else BooleanDistrib(0.05);
random Boolean MaryCalls ~
  if Alarm then BooleanDistrib(0.7)
  else BooleanDistrib(0.01);
```

Markov Random Field



Weighted Model Counting (WMC)

- Generalises propositional model counting (#SAT)
- Applications:
  - graphical models
  - probabilistic programming
  - neural-symbolic artificial intelligence
- Main types of algorithms:
  - using knowledge compilation
  - using a SAT solver
  - manipulating pseudo-Boolean functions

Example

$w(x) = 0.3,\ w(\neg x) = 0.7,$
$w(y) = 0.2,\ w(\neg y) = 0.8$

$\mathrm{WMC}(x \lor y) = w(x)w(y) + w(x)w(\neg y) + w(\neg x)w(y) = 0.44$

An Alternative Way to Think About WMC

- Let V be the set of variables.
- Then 2^{2^V} is the Boolean algebra of propositional formulas.

Definition

A measure is a function $\mu \colon 2^{2^V} \to \mathbb{R}_{\geq 0}$ such that:

- $\mu(\bot) = 0$;
- $\mu(x \vee y) = \mu(x) + \mu(y)$ whenever $x \wedge y = \bot$.

Observation

WMC corresponds to the process of calculating the value of $\mu(x)$ for some $x \in 2^{2^V}$.

The Limitations and Capabilities of Classical WMC

Observation

Classical WMC is only able to evaluate factorable measures (c.f., a collection of mutually independent random variables).

Theorem (Informal Version)

It is always possible to add more variables to turn a non-factorable measure into a factorable measure.

However, that is not necessarily a good idea!

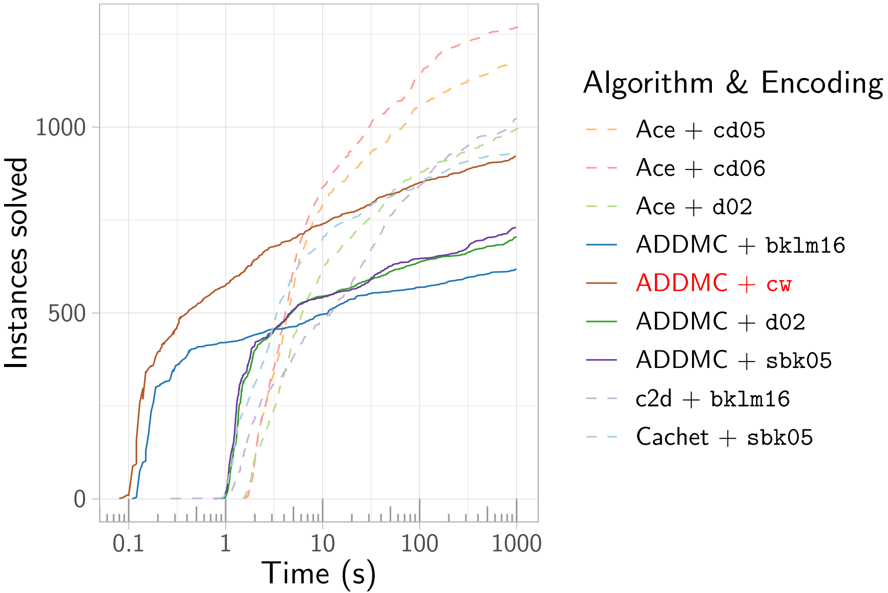
Conditional Probability Tables (CPTs) as Functions

- One variable for every random variable with two values.
- k variables for every random variable with $k > 2$ values.
- Define indicator functions of the form $[x]\colon 2^{\{x\}} \to \{0, 1\}$.
  - $[x](\emptyset) = 0$;
  - $[x](\{x\}) = 1$.
- Define $+$, $\cdot$, and scalar multiplication pointwise.
- Then a CPT can be represented as a function.

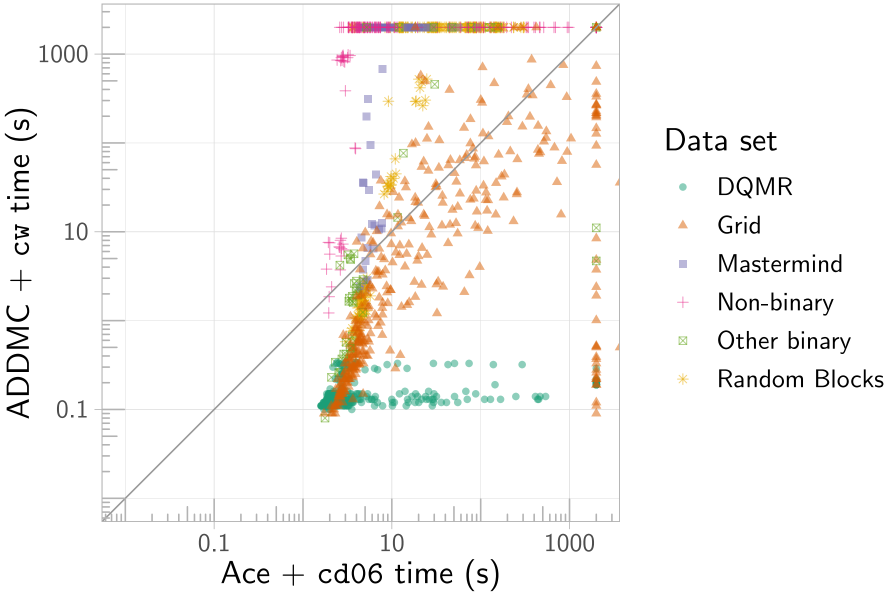
a	b	$\Pr(A = a \mid B = b)$
1	1	0.6
1	0	0.4
0	1	0.1
0	0	0.9

$$\begin{aligned}\mathrm{CPT_A} = {} & 0.6[\lambda_{A=1}] \cdot [\lambda_{B=1}] \\ & + 0.4[\lambda_{A=1}] \cdot \overline{[\lambda_{B=1}]} \\ & + 0.1\overline{[\lambda_{A=1}]} \cdot [\lambda_{B=1}] \\ & + 0.9\overline{[\lambda_{A=1}]} \cdot \overline{[\lambda_{B=1}]},\end{aligned}$$

Experimental Results



Comparison With the State of the Art



Summary and Future Work

- Classical WMC can represent any probability distribution by adding more variables.
- But this is not the right approach for WMC algorithms that support working directly with functions.
- Specifically with ADDMC, avoiding redundant variables resulted in 127 times faster inference.
- Could this idea be successfully applied to other applications of WMC or, perhaps, other WMC algorithms?
- Potential improvements to the encoding:
  - Apply ideas from other WMC encodings for Bayesian networks (e.g., prime implicants, log encoding).
  - Develop encoding tricks that apply to functions but not to conjunctive normal form.
  - More on this in our SAT 2021 paper *Weighted Model Counting Without Parameter Variables*.